Merge heißt Deploy

> Zielgruppe: Anwender und Betreuer · Stand: 06.11.2026

Ein Merge auf einen Stage-Branch ist kein Code-Review-Abschluss, sondern der Start eines Deployments. Je nach Pin der CI-Component wird sofort angewendet oder erst nach einem Klick. Und je nach Repo wandert der Commit danach weiter in die nächste Stage.

> Behandle jeden Merge auf `pre-development`, `development`, `staging` oder `production` wie ein Deployment. Sieh dir den Plan an, bevor du mergst.

Was ein Merge auslöst

Aktuelle Component (mit Freigabe-Gate):

1. `plan` plant und legt den Plan geschützt ab.
2. `apply-freigeben` wartet auf einen manuellen Klick.
3. `deploy` wendet genau diesen Plan an.

Ältere Pins (ohne Gate): Ein Merge auf `pre-development` plant **und wendet sofort an**, ohne Klick.

Promotion: Danach wandert der Commit nach den Regeln der Kaskade weiter. Bei älteren Pins ging das bis `development` und `staging` automatisch, z. B. bei den Breeze- und Freeze-Project-Deployern. Mit der aktuellen Component ist der Schritt nach `development` manuell.

Vorgehen vor einem Merge

1. Den Pin der Component im Repo prüfen: Gibt es `apply-freigeben`?
2. Den Plan gegen den Ziel-State ansehen (MR-Pipeline oder ein Lauf nur mit Plan).
3. Auf unerwartete Replaces achten. Drift durch neue Modul- oder Rollenversionen läuft im selben Apply mit.
4. Mehrere MRs im selben Repo nacheinander mergen, dazwischen rebasen und neu planen.
5. Bei Repos mit Promotion klären, ob die Änderung weiterwandern soll.

Stolperfallen

>

> Ein Merge, der „nur etwas hinzufügt“, kann trotzdem Replaces anwenden, wenn sich seit dem letzten Lauf Module oder Rollen geändert haben.

- `-o ci.skip` verhindert nur die Pipeline des Branch-Pushs, nicht die nach dem Merge.
- MR-Pipelines mit OpenBao-Anbindung planen nicht. Den Plan liefert dann erst die Pipeline auf dem Stage-Branch vor dem Gate.

Weiterführend

- Deployment-Repo mit der CI-Component
- Stages und Branches

Revision #1
Created 2026-11-06 18:10:45 UTC by Alexander Golembek
Updated 2026-11-06 18:10:46 UTC by Alexander Golembek